

Initiation au traitement du signal sous Matlab.

Fabien PIERRE

fabien.pierre@math.u-bordeaux1.fr

<http://sites.google.com/site/fabienpierre/enseignements>

Automne 2013.

Ce TP se veut être une introduction au logiciel Matlab, qui sera utilisé tout au long du Master TdSI.

La documentation de Matlab est très riche et complète. Il est vivement conseillé de la consulter souvent. Vous pouvez l'appeler à tout moment grâce à la commande `>> doc`. Si vous souhaitez une aide rapide sur une fonction particulière, vous pouvez également utiliser la commande suivante :

```
1 >> help nom_de_fonction
```

Internet constitue également une somme colossale d'informations et d'aides pour programmer en Matlab.

1 Affectation

- Affecter la valeur 3^4 à la variable a en affichant la valeur.
- Affecter la valeur $1.5^3 - 2.11^3$ à la variable b sans afficher le résultat.
- Afficher b .
- Calculer $a - b$ et afficher le résultat.
- Quelles sont les variables utilisées ? Quelle est la place mémoire utilisée ?

2 Vecteurs et matrices

- Construire le vecteur u des 50 premiers entiers pairs.
- Calculer $\|u\|^2$.
- Construire un vecteur aléatoire v de 50 éléments suivant une loi normale.
- Construire un vecteur aléatoire v de 50 éléments suivant une loi uniforme.
- Construire un vecteur aléatoire v de 50 éléments suivant une loi de Bernoulli de paramètre p .
- Construire une matrice B aléatoire de 50×50 éléments suivant une loi normale.
- Afficher l'histogramme des éléments de B .
- Calculer le produit v de la matrice B et du vecteur v .
- Construire la matrice composée des racines cubiques de tous les éléments de la matrice B .
- Construire la matrice C de taille 50×50 telle que $C(i, j) = B(i, j)^2$ si $B(i, j) < 0.5$, et $C(i, j) = B(i, j)$ sinon.
- Assembler la matrice M d'ordre n telle que sa première ligne et sa première colonne soit entièrement composé de 1 et que le sous-bloc d'ordre $n-1$ restant soit l'identité, pour $n = 10$ puis $n = 32$.
- Écrire une fonction Matlab prenant en entrée n et retournant la matrice M telle que $M_{ij} = i$ pour $(i, j) \in \{1..n\}^2$.

- Écrire une fonction Matlab prenant en entrée un vecteur v et un paramètre t et retournant son seuillage doux, i.e. le vecteur w tel que :

$$w_i = \begin{cases} v_i - t & \text{si } v_i > t, \\ v_i + t & \text{si } v_i < -t, \\ 0 & \text{sinon.} \end{cases} \quad (1)$$

3 Graphes (I)

Tracé de courbes en coordonnées cartésiennes : Sur l'intervalle $[-3;3]$, tracer les graphes de

$$f(x) = \sin(x) \quad (2)$$

et

$$g(x) = \frac{\sin(x)}{x} \quad (3)$$

(penser à la commande `plot`).

Tracé de courbes en coordonnées polaires : Tracer le graphe de

$$\rho = 3 \cos(2\theta) \quad (4)$$

(penser à la commande `polar`).

Graphique 3-D : Tracer sur $[-10, 10] \times [-10, 10]$ la fonction

$$h(x, y) = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}} \quad (5)$$

(penser aux commandes `meshgrid`, `mesh`, `surf`).

4 Graphes (II)

- Dessiner en coordonnées cartésiennes les fonctions :
 - $y = \sin(x)e^{-0.1x}/3$
 - $x(t) = \sin(t)$, $y(t) = -\sin(t)\cos(t)$
 - Placer des titres sur les figures précédentes, écrire les formules des fonctions sur les figures.
- Dessiner en coordonnées polaires les fonctions
 - $\rho = \theta + \frac{1}{\theta}$
 - $\rho = 2\frac{\sin(\theta)}{\theta}$
- Dessiner la surface $z = x^2 - \frac{1}{2y^2} - 6\sqrt{y}$
- Dessiner la surface de révolution engendrée par la fonction `sinh` (on utilisera la fonction `cylinder`).

5 Manipulation d'images.

Commencer par télécharger votre image préférée, et la déposer dans votre répertoire de travail. Vous pouvez utiliser l'image `lena`, par exemple disponible sur le lien http://www-2.cs.cmu.edu/~chuck/lennap/lena_std.tif.

Lecture. Pour utiliser une image sous Matlab, il faut commencer par la charger :

```
1 Im = imread('lena_std.tif'); % Lecture de l'image.
```

On travaillera souvent en image en niveaux de gris, pour cela, il faut transformer notre image couleur en niveau de gris. Observer la dimension `whos` ou `size` des objets manipulés :

```
1 Im = rgb2gray(Im);
```

Pour travailler avec, il faut la transformer au format double :

```
1 Im = double(Im); % Formatage.
```

Pour afficher une image en niveau de gris, vous pouvez utiliser le code suivant :

```
1 figure
2 imagesc(Im)
3 colormap gray
4 axis off
5 title('Mon titre.')
```

Pour éviter d'avoir à effectuer ces manipulations systématiquement, vous pouvez sauvegarder la matrice `Im` avec la commande `save('lena.mat', 'Im')`. Pour la récupérer utilisez la commande `load lena`.

Observer les effets des commandes `clc`, `clear all`, `close all`. Il est préférable de commencer systématiquement vos scripts par cette ligne de commande.

L'image considérée est entre 0 et 255.

– Afficher l'image inverse, i.e. l'image I telle que

$$I_{i,j} = 255 - Im_{i,j}.$$

- Afficher les valeurs minimales et maximales de l'image. Afficher leurs coordonnées.
- Afficher les moyennes et médianes de l'image.
- Afficher l'image, et le double de l'image. Commentez.
- Afficher l'image quantifiée sur 2,3,5 et 10 niveaux, i.e. l'image ne contenant plus que 2,3,5 et 10 valeurs possibles au lieu des 256 habituelles. (penser à la commande `floor`)

Enregistrement d'une image. Lorsque vous avez fini votre travail et que vous souhaitez sauvegarder une image, quelques précautions sont indispensables : il faut réajuster à l'échelle, et reformatter. En vous inspirant des lignes de code suivantes, écrivez une fonction Matlab qui prend en entrée une variable image, et une chaîne de caractères et qui sauvegarde l'image sous le nom donné par la chaîne de caractères.

```
1 Im = ( Im - min(Im(:)) ) / ( max(Im(:)) - min(Im(:)) ) * 255 ; % Mise a ...
    l'echelle.
2 imwrite( uint8(Im) , 'nom_de_l_image.png', 'png' ); % ...
    Enregistrement.
```

6 Filtre de Canny.

Le filtre de Canny permet de révéler les contours visibles sur une image.

Lissage gaussien. Le filtre de Canny est sensible au bruit. On va donc filtrer l'image par un noyau gaussien. On va effectuer la convolution d'une image par ce filtre (ces notions théoriques seront vues en cours ultérieurement).

Le filtre gaussien est une version discrétisée de la fonction

$$\frac{e^{-\frac{x^2 + y^2}{2\sigma^2}}}{\sqrt{2\pi}\sigma},$$

sur l'intervalle $[-3\sigma, 3\sigma]$.

- Construire un tel filtre.
- Appliquez le sur l'image `lena` (commande `conv2`). Explorer les options de la commande `conv2`. Trouvez la plus appropriée.
- Commentez. (On appelle cette opération un lissage gaussien, ou flou gaussien.)

Il s'agit maintenant de trouver les contours de l'image. On approxime le gradient ∇u de l'image par convolution avec le filtre $d_x = [-1, 1]$ pour la dérivée dans la direction horizontale, et $d_y = d_x^t$ pour la dérivée verticale.

Les contours sont donnés par les valeurs de $\sqrt{\frac{\partial u^2}{\partial x} + \frac{\partial u^2}{\partial y}}$ supérieure à un seuil donné.

- Calculer et afficher les contours.
- Commentez.

On veut tester la robustesse au bruit de cet algorithme. Pour cela on crée une image bruitée par un bruit gaussien centré. Un bruit gaussien est la réalisation d'une suite de variables aléatoires indépendantes de loi normale centrée. La variance s'appelle aussi *puissance* du bruit.

- Utilisez la commande `randn` pour ajouter un tel bruit.
- Affichez l'histogramme normalisé du bruit. Superposez-le à la courbe de densité de la loi normale (commande `normpdf`).
- Affichez l'image bruitée. Commentez. Faites varier la puissance du bruit.
- Appliquer le filtrage gaussien sur l'image bruitée.
- Réalisez une détection de contours sur l'image bruitée et sur l'image lissée.